PGPUB-DOCUMENT-NUMBER: 20020091670

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20020091670 A1

TITLE: Write anywhere file-system layout

PUBLICATION-DATE: July 11, 2002

US-CL-CURRENT: 707/1

APPL-NO: 09/ 954522

DATE FILED: September 11, 2001

RELATED-US-APPL-DATA:

child 09954522 A1 20010911 parent continuation-of 09153094 19980914 US GRANTED parent-patent 6289356 US child 09153094 19980914 US parent continuation-of 09108022 19980630 US GRANTED parent-patent 5963962 US child 09108022 19980630 US parent continuation-of 08454921 19950531 US GRANTED parent-patent 5819292 US child 08454921 19950531 US parent continuation-of 08071643 19930603 US ABANDONED

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	DOC-ID	APPL-DATE
US	PCT/US94/06320	1994US-PCT/US94/06320	June 2, 1994

---------- KWIC ---------

Summary of Invention Paragraph - BSTX:

[0025] The present invention also creates snapshots, which are virtual
read-only copies of the file system. A snapshot uses no disk space when it is
initially created. It is designed so that many different snapshots can be
created for the same file system. Unlike prior art file systems that create a
clone by duplicating the entire inode file and all of the indirect blocks, the
present invention duplicates only the inode that describes the inode file.
Thus, the actual disk space required for a snapshot is only the 128 bytes used
to store the duplicated inode. The 128 bytes of the present invention required
for a snapshot is significantly less than the many megabytes used for a clone
in the prior art.

Detail Description Paragraph - DETX:

[0127] WAFL supports up to 20 different snapshots that are numbered 1 through
20. Thus, WAFL allows the creation of multiple "clones" of the same file
system. Each snapshot is represented by a snapshot inode that is similar to

the representation of the active file system by a root inode.
Snapshots are
created by duplicating the root data structure of the file system.
In the
preferred embodiment, the root data structure is the root inode.
However, any
data structure representative of an entire file system could be
used. The
snapshot inodes reside in a fixed location in the inode file. The
limit of 20
snapshots is imposed by the size of the blkmap entries. WAFL
requires two
steps to create a new snapshot N: copy the root inode into the
inode for
snapshot N; and, copy bit 0 into bit N of each blkmap entry in the
blkmap file.
Bit 0 indicates the blocks that are referenced by the tree beneath
the root
inode.

Detail Description Paragraph - DETX:

[0155] The present invention limits the total number of snapshots
and keeps a
blkmap file that has entries with multiple bits for tracking the
snapshots
instead of using pointers having a COW bit as in Episode. An
unused block has
all zeroes for the bits in its blkmap file entry. Over time, the BITO
for the
active file system is usually turned on at some instant. Setting
BITO
identifies the corresponding block as allocated in the active file
system. As
indicated above, all snapshot bits are initially set to zero. If the

active file bit is cleared before any snapshot bits are set, the block is not present in any snapshot stored on disk. Therefore, the block is immediately available for reallocation and cannot be recovered subsequently from a snapshot.

Detail Description Paragraph - DETX:

[0168] In general, data on disk is not overwritten in the WAFL file system so as to protect data stored on disk. The only exception to this rule is atime overwrites for an inode as illustrated in FIGS. 23A-23B: When an "atime overwrites" occurs, the only data that is modified in a block of the inode file is the atime of one or more of the inodes it contains and the block is rewritten in the same location. This is the only exception in the WAFL system; otherwise new data is always written to new disk locations.